

Amendments to the Claims

Kindly cancel claims 2, 12, 19-30, and 39-42; add claims 43-70; and amend claims 1, 3, 4, 11, 13, 16, 31, 32, and 34 as follows:

1. (currently amended) A branch target address cache (BTAC) for providing a speculative target address to address selection logic, the address selection logic selecting a fetch address for addressing a line in an instruction cache, the BTAC providing the speculative target address based on a presumption that a branch instruction is present in the cache line, the BTAC comprising:
 - an array of storage elements, configured to cache target addresses of previously executed branch instructions and to store speculative branch information associated with said previously executed branch instructions, wherein said speculative branch information comprises a length of the branch instruction presumed present in the cache line;
 - an input, coupled to said array, for receiving the fetch address, to index into said array of storage elements to select one of said target addresses; and
 - an output, coupled to said array, for providing said one of said target addresses indexed by the fetch address to the address selection logic;
 - wherein said output provides said one of said target addresses to the address selection logic for selection as a subsequent fetch address whether or not a branch instruction is present in the line of the instruction cache addressed by the fetch address.
2. (canceled)
3. (currently amended) The branch target address cache of claim—2_1, further comprising:
 - a second output, coupled to said array, for providing a portion of said speculative branch information to control logic for controlling the address selection logic in response to said portion of said speculative branch information.
4. (currently amended) The branch target address cache of claim—2_1, wherein said speculative branch information comprises information predicting whether the branch instruction presumed present in the cache line will be taken.

5. (original) The branch target address cache of claim 4, wherein said information predicting whether the presumed branch instruction will be taken comprises a taken/not taken bit.
6. (original) The branch target address cache of claim 4, wherein said information predicting whether the presumed branch instruction will be taken comprises a plurality of bits.
7. (original) The branch target address cache of claim 6, wherein said plurality of bits is stored in a saturating up/down counter.
8. (original) The branch target address cache of claim 3, wherein said portion of said speculative branch information comprises an indication of whether said one of said target addresses is a valid target address.
9. (original) The branch target address cache of claim 8, wherein said indication is populated to indicate said one of said target addresses is a valid target address in response to execution of the presumed branch instruction, wherein said one of said target addresses is resolved.
10. (original) The branch target address cache of claim 8, wherein said indication is populated to indicate said one of said target addresses is not a valid target address in response to detecting said one of said target addresses is erroneous subsequent to said providing said one of said target addresses on said output.
11. (currently amended) The branch target address cache of claim ~~2~~ 1, wherein said speculative branch information comprises information specifying a location within the cache line of the branch instruction presumed present in the cache line.
12. (canceled)
13. (currently amended) The branch target address cache of claim ~~2~~ 1, wherein said speculative branch information comprises an indication of a type of the branch instruction presumed present in the cache line.
14. (original) The branch target address cache of claim 13, wherein said indication of said type of the branch instruction indicates whether the branch instruction is a call instruction.

15. (original) The branch target address cache of claim 13, wherein said indication of said type of the branch instruction indicates whether the branch instruction is a return instruction.
16. (currently amended) The branch target address cache of claim ~~2~~ 1, wherein said speculative branch information comprises an indication of whether the branch instruction presumed present in the cache line spans more than one line in the instruction cache.
17. (original) The branch target address cache of claim 1, wherein each of said storage elements is configured to cache a plurality of target addresses.
18. (original) The branch target address cache of claim 1, wherein the branch target address cache is external to the instruction cache.
19. (canceled)
20. (canceled)
21. (canceled)
22. (canceled)
23. (canceled)
24. (canceled)
25. (canceled)
26. (canceled)
27. (canceled)
28. (canceled)
29. (canceled)
30. (canceled)

31. (currently amended) A pipelined microprocessor comprising:
- an instruction cache that is indexed by a fetch address, said instruction cache for caching instructions, and for providing said instructions to an instruction buffer;
 - a branch target address cache, coupled to said instruction buffer and indexed by said fetch address, for caching branch target addresses of previously executed branch instructions;
 - said instruction buffer comprising a plurality of ~~hit~~-indicators that are associated with said instructions, said indicators specifying whether the microprocessor has speculatively branched to one of said branch target addresses.
32. (currently amended) The microprocessor of claim 31, wherein said instruction buffer includes one of said plurality of ~~hit~~-indicators associated with each byte of said each of said instructions stored in said instruction buffer.
33. (original) The microprocessor of claim 31, wherein said instruction cache and said branch target address cache are accessed substantially in parallel.

34. (currently amended) A method of speculatively branching in a pipelined microprocessor, comprising:

caching a plurality of branch target addresses of previously executed branch instructions and an indication of whether each of the branch instructions spans an instruction cache line in a branch target address cache (BTAC);

accessing said BTAC with a fetch address of an instruction cache after said caching;

determining whether said fetch address hits in said BTAC in response to said accessing; and

branching the microprocessor to one of said plurality of branch target addresses selected by said fetch address if said fetch address hits in said BTAC whether or not a branch instruction is cached in a line of said instruction cache indexed by said fetch address.

35. (original) The method of claim 34, further comprising:

storing a branch direction prediction associated with each of said plurality of branch target addresses prior to said accessing said BTAC.

36. (original) The method of claim 35, wherein said branching the microprocessor to said one of said plurality of branch target addresses selected by said fetch address is performed only if said associated branch direction prediction indicates said branch instruction will be taken.

37. (original) The method of claim 34, further comprising:

storing an indication said branching was performed if said branching is performed.

38. (original) The method of claim 37, wherein said storing said indication said branching was performed comprises storing said indication in an instruction buffer.

39. (canceled)

40. (canceled)

41. (canceled)

42. (canceled)

43. (new) The microprocessor of claim 32, further comprising:

instruction decode logic, coupled to said instruction buffer, for decoding said instructions to indicate which byte of each of said instructions is an opcode byte; and

prediction check logic, coupled to receive said plurality of indicators from said instruction buffer, wherein if one of said plurality of indicators associated with one of said instructions specifies the microprocessor has speculatively branched to said one of said branch target addresses of said one of said instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if a same byte of said one of said instructions is not indicated by said instruction decode logic and by said plurality of indicators stored in said instruction buffer to be said opcode byte.

44. (new) The microprocessor of claim 31, further comprising:

instruction decode logic, coupled to said instruction buffer, for decoding said instructions to indicate whether each of said instructions is a non-branch instruction; and

prediction check logic, coupled to receive said plurality of indicators from said instruction buffer, wherein if one of said plurality of indicators associated with one of said instructions specifies the microprocessor has speculatively branched to said one of said branch target addresses of said one of said instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if said instruction decode logic indicates said one of said instructions is a non-branch instruction.

45. (new) The microprocessor of claim 31, further comprising:
instruction decode logic, coupled to said instruction buffer, for determining a length of each of said instructions; and
prediction check logic, coupled to receive said plurality of indicators from said instruction buffer, wherein if one of said plurality of indicators associated with one of said instructions specifies the microprocessor has speculatively branched to said one of said branch target addresses of said one of said instructions, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if said length received from said instruction decode logic does not match a speculative length of said one of said instructions provided by said branch target address cache.
46. (new) The microprocessor of claim 31, further comprising:
prediction check logic, coupled to receive said plurality of indicators from said instruction buffer, wherein said one of said instructions is a branch instruction, wherein if one of said plurality of indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if a resolved direction of said branch instruction does not match a direction of said branch instruction predicted by said branch target address cache.

47. (new) The microprocessor of claim 31, further comprising:
prediction check logic, coupled to receive said plurality of indicators from said instruction buffer, wherein said one of said instructions is a branch instruction, wherein if one of said plurality of indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said prediction check logic indicates that the microprocessor erroneously branched to said one of said branch target addresses if a resolved target address of said branch instruction does not match said one of said branch target addresses to which the microprocessor speculatively branched.
48. (new) The microprocessor of claim 31, further comprising:
a non-speculative branch predictor, coupled to instruction buffer, for generating a non-speculative predicted target address of a branch instruction for which said branch target address cache provided said one of said branch target addresses to which the microprocessor speculatively branched; and
branch control logic, coupled to receive said plurality of indicators from said instruction buffer, wherein if one of said plurality of indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said branch control logic causes the microprocessor to branch to said non-speculative predicted target address if said non-speculative predicted target address generated by said non-speculative branch predictor does not match said one of said branch target addresses of said branch instruction provided by said branch target address cache.

49. (new) The microprocessor of claim 31, further comprising:
- a non-speculative branch predictor, coupled to instruction buffer, for generating a non-speculative predicted direction of a branch instruction for which said branch target address cache provided said one of said branch target addresses to which the microprocessor speculatively branched; and
 - branch control logic, coupled to receive said plurality of indicators from said instruction buffer, wherein if one of said plurality of indicators associated with said branch instruction specifies the microprocessor has speculatively branched to said one of said branch target addresses of said branch instruction, said branch control logic causes the microprocessor to branch to a next instruction sequential to said branch instruction if said non-speculative predicted direction generated by said non-speculative branch predictor is a not taken prediction.

50. (new) A branch target address cache (BTAC) for providing a speculative target address to address selection logic, the address selection logic selecting a fetch address for addressing a line in an instruction cache, the BTAC providing the speculative target address based on a presumption that a branch instruction is present in the cache line, the BTAC comprising:
- an array of storage elements, configured to cache target addresses of previously executed branch instructions and to store speculative branch information associated with said previously executed branch instructions, wherein said speculative branch information comprises an indication of whether the branch instruction presumed present in the cache line spans more than one line in the instruction cache;
 - an input, coupled to said array, for receiving the fetch address, to index into said array of storage elements to select one of said target addresses; and
 - an output, coupled to said array, for providing said one of said target addresses indexed by the fetch address to the address selection logic;
- wherein said output provides said one of said target addresses to the address selection logic for selection as a subsequent fetch address whether or not a branch instruction is present in the line of the instruction cache addressed by the fetch address.
51. (new) The branch target address cache of claim 50, further comprising:
- a second output, coupled to said array, for providing a portion of said speculative branch information to control logic for controlling the address selection logic in response to said portion of said speculative branch information.
52. (new) The branch target address cache of claim 50, wherein said speculative branch information comprises information predicting whether the branch instruction presumed present in the cache line will be taken.
53. (new) The branch target address cache of claim 52, wherein said information predicting whether the presumed branch instruction will be taken comprises a taken/not taken bit.

54. (new) The branch target address cache of claim 52, wherein said information predicting whether the presumed branch instruction will be taken comprises a plurality of bits.
55. (new) The branch target address cache of claim 54, wherein said plurality of bits is stored in a saturating up/down counter.
56. (new) The branch target address cache of claim 51, wherein said portion of said speculative branch information comprises an indication of whether said one of said target addresses is a valid target address.
57. (new) The branch target address cache of claim 56, wherein said indication is populated to indicate said one of said target addresses is a valid target address in response to execution of the presumed branch instruction, wherein said one of said target addresses is resolved.
58. (new) The branch target address cache of claim 56, wherein said indication is populated to indicate said one of said target addresses is not a valid target address in response to detecting said one of said target addresses is erroneous subsequent to said providing said one of said target addresses on said output.
59. (new) The branch target address cache of claim 50, wherein said speculative branch information comprises information specifying a location within the cache line of the branch instruction presumed present in the cache line.
60. (new) The branch target address cache of claim 50, wherein said speculative branch information comprises a length of the branch instruction presumed present in the cache line.
61. (new) The branch target address cache of claim 50, wherein said speculative branch information comprises an indication of a type of the branch instruction presumed present in the cache line.
62. (new) The branch target address cache of claim 61, wherein said indication of said type of the branch instruction indicates whether the branch instruction is a call instruction.
63. (new) The branch target address cache of claim 61, wherein said indication of said type of the branch instruction indicates whether the branch instruction is a return instruction.

64. (new) The branch target address cache of claim 50, wherein each of said storage elements is configured to cache a plurality of target addresses.
65. (new) The branch target address cache of claim 50, wherein the branch target address cache is external to the instruction cache.

66. (new) A method of speculatively branching in a pipelined microprocessor, comprising:
- caching a plurality of branch target addresses of previously executed branch instructions and a length of each of the branch instructions in a branch target address cache (BTAC);
 - accessing said BTAC with a fetch address of an instruction cache after said caching;
 - determining whether said fetch address hits in said BTAC in response to said accessing; and
 - branching the microprocessor to one of said plurality of branch target addresses selected by said fetch address if said fetch address hits in said BTAC whether or not a branch instruction is cached in a line of said instruction cache indexed by said fetch address.
67. (new) The method of claim 66, further comprising:
- storing a branch direction prediction associated with each of said plurality of branch target addresses prior to said accessing said BTAC.
68. (new) The method of claim 67, wherein said branching the microprocessor to said one of said plurality of branch target addresses selected by said fetch address is performed only if said associated branch direction prediction indicates said branch instruction will be taken.
69. (new) The method of claim 66, further comprising:
- storing an indication said branching was performed if said branching is performed.
70. (new) The method of claim 69, wherein said storing said indication said branching was performed comprises storing said indication in an instruction buffer.